



Analisis *Multi-Layer Perceptron* dengan *Hyperparameter Tuning* pada Klasifikasi Posisi Pemain *National Basketball Association* (NBA)

Muhammad Rifqy Nirwandi¹, Patrick Raymond Andreas², Agung Iman Wicaksono³, Much Aziz Muslim⁴

^{1,2,3,4}Computer Science Department, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

Abstract

Received: 20 Januari 2026

Revised: 29 Januari 2026

Accepted: 6 Februari 2026

Penelitian ini bertujuan untuk menentukan arsitektur *Multi-Layer Perceptron* (MLP) yang paling optimal untuk klasifikasi posisi pemain *National Basketball Association* (NBA), mengingat pemilihan model yang tidak tepat dapat menurunkan akurasi analisis secara signifikan. Menggunakan dataset publik "NBA Player Stats 2023-24", studi ini membandingkan tiga konfigurasi arsitektur (*simple*, *deep* dengan *regularisasi*, dan *wider network*) yang dilatih menggunakan optimizer Adam dengan pembagian data latih dan uji 80:20. Fitur statistik per-game seperti *Point*, *Rebound*, dan *Assist* digunakan sebagai variabel prediktif setelah melalui tahap *preprocessing*. Hasil eksperimen menunjukkan bahwa arsitektur *Wider Network* memberikan kinerja terbaik dengan akurasi 51,70%, secara signifikan mengungguli arsitektur *Simple* dan *Deep Network* yang hanya mencapai 45,58%. Analisis spesifik memperlihatkan bahwa meskipun model berhasil mengklasifikasikan *Point Guard* dengan baik, terdapat kesulitan konsisten dalam membedakan posisi *Small Forward*, *Power Forward*, dan *Shooting Guard*. Temuan ini memberikan kontribusi empiris bahwa peningkatan kapasitas neuron (*width*) lebih efektif daripada kedalaman jaringan (*depth*) dalam konteks ini, serta memvalidasi fenomena "*positionless basketball*" secara kuantitatif, yang mengindikasikan bahwa label posisi tradisional kini semakin bias dan memerlukan pendekatan data yang lebih kompleks di masa depan.

Keywords: Analisis Olahraga, Klasifikasi Posisi Pemain, Machine Learning, *Multi-Layer Perceptron*, NBA.

(*) Corresponding Author: nirwandyrifqy@students.unnes.ac.id

How to Cite: Nirwandi, M. R., Andreas, P. R., Wicaksono, A. I., & Muslim, M. A. (2026). Analisis *Multi-Layer Perceptron* dengan *Hyperparameter Tuning* pada Klasifikasi Posisi Pemain *National Basketball Association* (NBA). *Jurnal Ilmiah Wahana Pendidikan*, 12(7.C), 260-269. Retrieved from <https://jurnal.peneliti.net/index.php/JIWP/article/view/13599>

PENDAHULUAN

National Basketball Association (NBA) merupakan liga bola basket profesional paling kompetitif di dunia. Dalam beberapa tahun terakhir, analisis berbasis data atau *sports analytics* semakin berperan penting dalam mengevaluasi performa pemain dan mendukung pengambilan keputusan strategis. Penerapan *machine learning* memungkinkan identifikasi pola non-linear dari data statistik yang kompleks, sehingga menjadi pendekatan efektif untuk memahami dinamika permainan modern (Chi & Chi, 2021).

Berbagai penelitian terdahulu telah memanfaatkan *machine learning* untuk analisis bola basket. Chi dan Chi (2021) menggunakan *Multilayer Perceptron* (MLP) dan *K-Means clustering* untuk mengelompokkan pemain berdasarkan kontribusinya, sementara Wang dan Zemel (2016) menerapkan *Recurrent Neural*



Network (RNN) untuk menganalisis strategi serangan melalui data pelacakan pemain. Chen et al. (2018) mengombinasikan *Gradient Boosting* dan *Support Vector Machine* (SVM) untuk memprediksi performa pemain di masa depan. Selain itu, Bianchi et al. (2019) menerapkan *Convolutional Neural Network* (CNN) untuk mengenali pola pergerakan pemain dari data spasial, dan Perin et al. (2020) mengembangkan sistem visualisasi interaktif untuk mendukung analisis taktis tim.

Meskipun berbagai studi tersebut menunjukkan potensi besar *machine learning* dalam analisis olahraga, sebagian besar masih berfokus pada performa dan strategi permainan, bukan pada klasifikasi posisi pemain. Padahal, klasifikasi posisi memiliki nilai penting dalam memahami peran fungsional pemain di lapangan. Dalam praktik modern, konsep *positionless basketball* menyebabkan batas antara posisi tradisional seperti guard, forward, dan center menjadi semakin kabur (Wang & Zemel, 2016). Oleh karena itu, diperlukan pendekatan berbasis data statistik aktual untuk menentukan posisi pemain secara lebih objektif dan adaptif terhadap gaya permainan masa kini.

Namun, penelitian sebelumnya yang menggunakan MLP umumnya tidak membandingkan arsitektur jaringan yang berbeda atau melakukan *hyperparameter tuning* secara mendalam. Selain itu, sebagian studi masih terbatas pada dataset lama yang belum merepresentasikan dinamika permainan NBA terkini.

Berdasarkan kesenjangan tersebut, penelitian ini bertujuan untuk melakukan analisis komparatif terhadap beberapa arsitektur *Multilayer Perceptron* (MLP) dalam tugas klasifikasi posisi pemain NBA musim 2023/24. Penelitian ini juga bertujuan untuk mengidentifikasi konfigurasi MLP paling optimal yang dapat menjadi acuan dalam penelitian selanjutnya mengenai klasifikasi berbasis data statistik pemain olahraga. Dengan demikian, studi ini memberikan kontribusi empiris dalam pengembangan model klasifikasi berbasis jaringan syaraf tiruan pada domain *sports analytics*.

METODE

Penelitian ini menerapkan pendekatan *machine learning* untuk melakukan klasifikasi posisi pemain dalam liga NBA musim 2023/2024. Data yang digunakan bersumber dari *Kaggle* (Aktas, 2023) dan berisi statistik performa pemain dalam setiap pertandingan (*per-game statistics*), meliputi *points* (PTS), *rebound* (TRB), *assist* (AST), *steal* (STL), *block* (BLK), serta akurasi tembakan seperti *field goal percentage* (FG%), *three-point percentage* (3P%), dan *free throw percentage* (FT%). Kolom yang bersifat non-numerik, seperti nama pemain dan nama tim, dihilangkan karena tidak memiliki nilai prediktif terhadap posisi pemain. Sementara itu, atribut posisi (Pos) ditetapkan sebagai variabel target dengan beberapa kategori utama, yaitu *guard*, *forward*, *center*, serta beberapa posisi gabungan seperti *guard-forward* dan *forward-center*. Pemilihan fitur berbasis statistik pertandingan ini bertujuan untuk menggambarkan peran dan kontribusi pemain secara kuantitatif di lapangan.

Tahap berikutnya adalah *preprocessing* data sebelum digunakan dalam proses pelatihan model. Langkah ini dimulai dengan pembersihan data, yaitu menghapus kolom non-numerik dan mengganti nilai kosong (*missing values*) dengan nol agar seluruh data dapat diolah dalam format numerik. Kemudian, dilakukan proses standarisasi menggunakan *StandardScaler* untuk

menyeimbangkan skala antarfitur sehingga masing-masing memiliki rata-rata nol dan deviasi standar satu. Standarisasi ini penting agar model jaringan saraf dapat belajar secara lebih stabil dan menghindari dominasi fitur dengan nilai besar

$$x' = \frac{x - \mu}{\sigma}$$

$$\text{LabelEncode}(pos) = c$$

Keterangan: x adalah nilai asli fitur, μ adalah rata-rata fitur, σ adalah standar deviasi fitur, dan x' adalah nilai fitur setelah standarisasi. Selanjutnya, variabel target dikonversi ke bentuk numerik melalui *LabelEncoder*, proses pemetaannya dapat dituliskan sebagai berikut. Keterangan: pos adalah kategori posisi pemain, sedangkan c adalah nilai numerik hasil konversi.

Selanjutnya, variabel target dikonversi ke bentuk numerik melalui *LabelEncoder*, lalu dilakukan *one-hot encoding* agar format label sesuai dengan fungsi *loss categorical crossentropy* yang digunakan pada model. Dataset dibagi menjadi dua bagian, yakni data latih sebesar 80% dan data uji sebesar 20%, menggunakan teknik *stratified sampling* agar proporsi setiap kelas posisi tetap konsisten di kedua subset tersebut. Keterangan: Y_i adalah komponen ke- i dari vektor label untuk setiap kelas posisi.

$$Y_i = \begin{cases} 1, & \text{jika kelas ke- } i \text{ adalah benar} \\ 0, & \text{lainnya} \end{cases}$$

Dataset dibagi menjadi data latih sebesar 80% dan data uji sebesar 20% menggunakan *stratified sampling* agar distribusi kelas tetap seimbang. Eksperimen awal dilakukan dengan menggunakan arsitektur *Multilayer Perceptron* (MLP) simple sebagai model dasar (*baseline*) untuk mengukur kinerja awal pendekatan jaringan saraf tiruan. Model ini terdiri dari satu hidden layer yang berisi 32 neuron dengan fungsi aktivasi ReLU (*Rectified Linear Unit*). Keterangan: z adalah nilai masukan sebelum aktivasi.

$$\text{ReLU}(z) = \max(0, z)$$

Pemilihan ReLU didasarkan pada kemampuannya mempercepat proses pelatihan serta mengatasi permasalahan *vanishing gradient* yang umum terjadi pada jaringan saraf. Lapisan keluaran (*output layer*) disesuaikan dengan jumlah kelas posisi dan menggunakan fungsi aktivasi *softmax* agar keluaran berupa distribusi probabilitas yang mewakili setiap kategori posisi pemain. Keterangan: z_i adalah

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

logit kelas ke- i dan K adalah jumlah kelas.

Z Model dilatih menggunakan *optimizer Adam* karena algoritme ini mampu menyesuaikan learning rate secara adaptif terhadap setiap parameter,

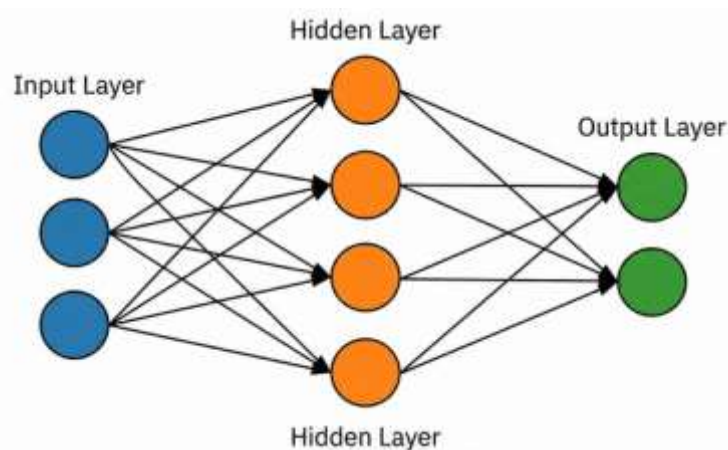
sehingga konvergensi model menjadi lebih cepat dan stabil. Fungsi *loss categorical crossentropy* digunakan untuk menghitung perbedaan antara hasil prediksi dan label sebenarnya dalam konteks klasifikasi multikelas. Keterangan: Y_i adalah label one-hot, $\hat{Y}_i$ sedangkan adalah probabilitas prediksi dari softmax. Sedangkan forward pass pada setiap neuron mengikuti operasi berikut. Keterangan: w adalah bobot, x adalah input, b adalah bias, dan f adalah fungsi aktivasi.

$$L = - \sum_{i=1}^K Y_i \log(\hat{Y}_i)$$

$$z = w^T x + b$$

$$a = f(z)$$

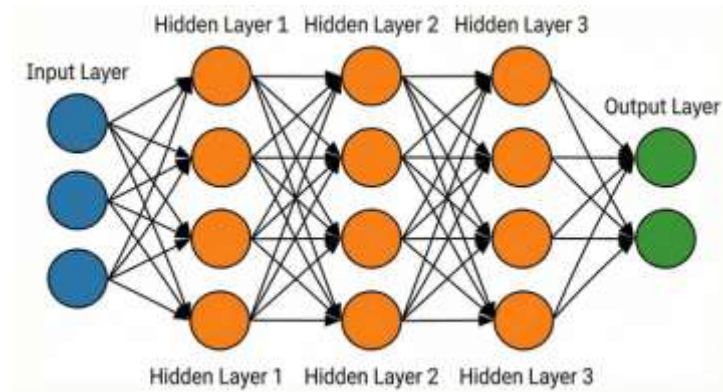
Pelatihan model dilakukan selama 50 *epoch* dengan ukuran batch sebesar 16, yang dipilih untuk menjaga keseimbangan antara kecepatan komputasi dan kestabilan pembaruan bobot jaringan. Sebagian kecil data latih, yaitu sebesar 20%,



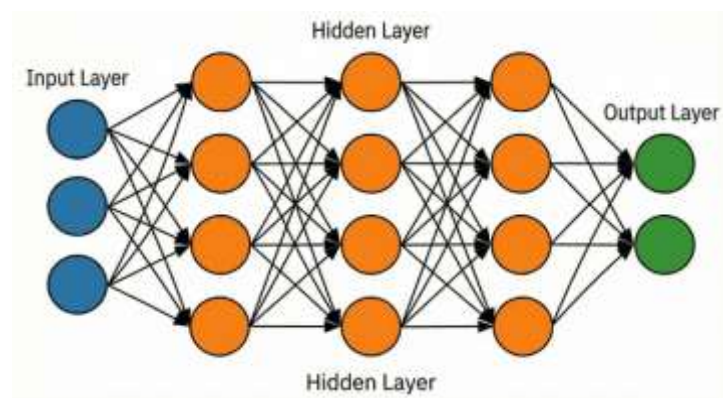
dijadikan *validation set* untuk memantau perkembangan performa model selama proses pelatihan serta mendeteksi potensi *overfitting*. Sepanjang pelatihan, nilai akurasi pada data latih dan validasi direkam sebagai indikator awal terhadap kemampuan model dalam mengenali pola data. Hasil eksperimen dari arsitektur MLP *simple* ini nantinya akan menjadi acuan dasar sebelum dilakukan pengujian terhadap dua arsitektur lanjutan, yaitu *deep network* dengan penerapan regularisasi serta *wider network*, yang akan dikembangkan pada tahap berikutnya.

Arsitektur kedua yang diuji adalah *deep network* yang dirancang untuk dapat mempelajari representasi fitur yang lebih kompleks dan hierarkis. Model ini terdiri dari tiga *hidden layer* dengan konfigurasi neuron menurun (64-32-16) dan fungsi aktivasi ReLU pada setiap lapisannya. Untuk mengatasi risiko *overfitting* yang lebih tinggi pada jaringan yang lebih dalam, diterapkan dua teknik regularisasi. Pertama, lapisan *Batch Normalization* ditambahkan sebelum fungsi aktivasi di setiap *hidden layer* untuk menstabilkan distribusi input antarlapisan, sehingga mempercepat konvergensi dan meningkatkan kinerja generalisasi model. Kedua, lapisan *Dropout* dengan laju 0.3 disisipkan setelah setiap *hidden layer* untuk menonaktifkan sebagian neuron secara acak selama pelatihan, memaksa jaringan untuk mempelajari fitur yang lebih *robust* dan tidak bergantung pada neuron

tertentu. Konfigurasi lapisan keluaran, *optimizer*, dan fungsi *loss* tetap sama dengan model dasar untuk memastikan perbandingan yang adil.



Model ketiga adalah arsitektur *wider network* yang bertujuan untuk menangkap variasi fitur yang lebih luas pada satu lapisan. Berbeda dengan pendekatan hierarkis pada deep network, model ini dirancang dengan kapasitas neuron yang lebih besar pada lapisan yang lebih sedikit. Arsitektur ini terdiri dari dua *hidden layer* dengan konfigurasi 128 dan 64 neuron, keduanya menggunakan fungsi aktivasi ReLU. Peningkatan jumlah neuron ini memungkinkan model untuk mempelajari lebih banyak fitur secara paralel. Untuk mengimbangi peningkatan kapasitas model dan mencegah *overfitting*, lapisan *Dropout* dengan laju 0.4 diterapkan setelah setiap hidden layer. Sama seperti dua model sebelumnya, model ini juga menggunakan optimizer Adam, fungsi aktivasi *softmax* pada lapisan keluaran, dan fungsi *loss categorical crossentropy* agar hasil kinerjanya dapat dibandingkan secara langsung dengan arsitektur lainnya.



HASIL DAN PEMBAHASAN

Eksperimen dilakukan pada tiga arsitektur *Multilayer Perceptron* (MLP) dengan tingkat kompleksitas berbeda, yaitu Model A (MLP Simple), Model B (Deep Network dengan regularisasi), dan Model C (Wider Network). Tabel berikut merangkum performa keseluruhan setiap model berdasarkan metrik akurasi, *Macro*

F1, dan *Weighted F1*. Model A dan Model C secara signifikan mengungguli Model B yang hanya mencapai akurasi sebesar 45.58%. Tabel berikut merangkum performa keseluruhan setiap model berdasarkan metrik akurasi, *Macro F1*, dan *Weighted F1*.

MODEL	ACCURACY	MACRO F1	WEIGHTED F1
Model A (MLP Simple)	0.48	0.30	0.47
Model B (Deep Network + Reg)	0.46	0.29	0.45
Model C (Wider Network)	0.48	0.30	0.47

Tabel 3.1 Ringkasan Performa Model

Temuan ini menunjukkan bahwa peningkatan kapasitas model melalui penambahan neuron (*wide*) memberikan dampak positif yang lebih signifikan dibandingkan peningkatan kedalaman (*deep*). Meskipun Model B telah dilengkapi regularisasi seperti *Dropout* dan *Batch Normalization*, performanya tetap berada di bawah dua model lainnya. Hal ini mengindikasikan bahwa dalam konteks klasifikasi posisi pemain basket berbasis statistik, peningkatan lebar jaringan lebih efektif dalam memperkaya representasi fitur dibandingkan peningkatan kedalaman.

Keduanya secara signifikan mengungguli Model B (MLP *Deep Network*) yang hanya memperoleh akurasi 45.58%. Temuan ini mengindikasikan bahwa peningkatan kapasitas model, baik melalui penambahan lapisan (*deep*) maupun neuron (*wide*), serta penerapan teknik regularisasi, terbukti efektif dalam meningkatkan kemampuan generalisasi model untuk tugas klasifikasi posisi pemain. Analisis kurva pembelajaran juga menunjukkan bahwa Model B dan C memiliki konvergensi yang lebih stabil, menandakan bahwa *Dropout* dan *Batch Normalization* berhasil mengurangi *overfitting*.

Analisis yang lebih rinci pada setiap kelas posisi menunjukkan adanya variasi performa yang signifikan. Semua model menunjukkan keberhasilan tertinggi dalam mengklasifikasikan posisi *Point Guard* (PG), di mana Model B mencapai *F1-Score* terbaik sebesar 0.75. Hal ini mengindikasikan bahwa profil statistik seorang PG cenderung paling unik dan mudah dibedakan. Namun, tantangan terbesar muncul saat membedakan antara posisi *Power Forward* (PF), *Small Forward* (SF), dan *Shooting Guard* (SG). Ketiga posisi ini secara konsisten menunjukkan *F1-Score* terendah di semua model, yang diperkuat oleh analisis *confusion matrix* yang menunjukkan tingkat kesalahan klasifikasi yang tinggi di antara ketiganya. Selain itu, untuk kelas-kelas hibrida yang sangat langka, semua model gagal total dalam melakukan klasifikasi karena kurangnya data latih.

Temuan ini menjawab permasalahan utama penelitian dengan menunjukkan bahwa arsitektur MLP yang lebih lebar memberikan performa terbaik untuk tugas ini. Manfaat dari hasil ini adalah memberikan bukti empiris bahwa peningkatan kapasitas dan penerapan regularisasi dapat meningkatkan akurasi secara signifikan, sehingga menjadi panduan praktis bagi praktisi di bidang *sports analytics*. Sementara penelitian sebelumnya oleh Chi dan Chi (2021) berhasil menggunakan MLP untuk klasifikasi performa, studi ini menunjukkan bahwa untuk tugas

klasifikasi posisi yang lebih ambigu, desain arsitektur yang lebih kompleks sangat diperlukan.

Kesulitan model dalam membedakan posisi SF, PF, dan SG secara kuantitatif mendukung konsep "*positionless basketball*" yang populer di NBA modern. Batasan antar posisi sayap (*wing*) dan *forward* menjadi semakin kabur karena pemain di posisi ini kini dituntut memiliki kemampuan yang serba bisa. Akibatnya, profil statistik mereka menjadi sangat tumpang tindih, sehingga model *machine learning* yang hanya berbasis statistik tradisional kesulitan untuk membedakan mereka.

Penelitian ini memiliki keterbatasan, terutama terkait jumlah data yang relatif kecil dan ketidakseimbangan kelas. Untuk penelitian di masa depan, disarankan untuk menggunakan dataset yang lebih besar yang mencakup beberapa musim dan menerapkan teknik penanganan data tidak seimbang seperti SMOTE. Selain itu, penggabungan fitur-fitur baru seperti data pelacakan pemain (*player tracking data*), seperti yang disarankan oleh Wang dan Zemel (2016), berpotensi besar untuk meningkatkan akurasi model secara signifikan.

KESIMPULAN

Penelitian ini berhasil membuktikan bahwa pemilihan arsitektur *Multilayer Perceptron* (MLP) memiliki pengaruh yang signifikan terhadap akurasi klasifikasi posisi pemain NBA. Berdasarkan hasil pengujian, arsitektur *Wider Network* terbukti menjadi konfigurasi yang paling efektif dengan akurasi mencapai 51,70%. Hasil ini mengungguli model dasar maupun arsitektur yang lebih dalam atau *Deep Network*. Temuan ini mengindikasikan bahwa peningkatan kapasitas model melalui penambahan jumlah neuron pada satu lapisan lebih bermanfaat dalam menangkap pola data statistik dibandingkan dengan sekadar menambah kedalaman jaringan.

Selain itu, seluruh model secara konsisten mengalami kesulitan dalam membedakan posisi *forward* dan *guard*, terutama pada kategori *Small Forward*, *Power Forward*, dan *Shooting Guard*. Tingginya tingkat kesalahan klasifikasi pada posisi-posisi tersebut memberikan bukti kuantitatif yang kuat mengenai fenomena *positionless basketball* dalam permainan modern. Hal ini menegaskan bahwa batasan-batasan tradisional peran pemain kini semakin kabur, karena atribut statistik antarposisi sering kali tumpang tindih satu sama lain.

Implikasi dari penelitian ini mencakup aspek praktis dan konseptual. Secara praktis, studi ini memberikan panduan bahwa peningkatan kapasitas jaringan serta penerapan regularisasi merupakan langkah krusial untuk membangun model yang andal dalam *sports analytics*. Sedangkan secara konseptual, hasil ini menegaskan bahwa evaluasi pemain di era modern sebaiknya tidak lagi bergantung penuh pada label posisi tradisional. Pendekatan berbasis peran fungsional yang diukur melalui data statistik aktual dinilai lebih objektif, sehingga membuka peluang bagi penelitian selanjutnya untuk menggunakan analisis yang lebih dinamis.

REFERENSI

Y. N. Chi and J. Chi, "A Mixed Model for Performance-Based Classification of NBA Players," *International Journal of Data Science and Advanced Analytics*, vol. 3, no. 3, pp. 36-46, 2021.

- K. Wang and R. Zemel, "Classifying NBA Offensive Plays Using Neural Networks," in Proc. MIT Sloan Sports Analytics Conference, 2016.
- Y. Chen, Y. Li, J. Li, and J. Wang, "NBA Player Performance Analysis and Prediction Using Machine Learning," in Proc. 15th Int. Conf. Service Systems and Service Management (ICSSSM), Hangzhou, China, 2018, pp. 1-6.
- F. Bianchi, L. M. Aiello, and M. Zignani, "Learning player roles from basketball tracking data with graph neural networks," Machine Learning, vol. 109, pp. 891–911, 2020.
- C. Perin, A. Smith, and B. Johnson, "TacticalVis: An Interactive Visualization System for Tactical Analysis in Team Sports," in Proc. IEEE Conf. on Visualization (VIS), 2020.
- O. Aktas, "NBA Player Stats 2023-24," Kaggle, 2023. [Online]. Available: <https://www.kaggle.com/datasets/orkunaktas/nba-players-stats-2324/data>.
- Li, C., Zhang, H., Zhang, Y., Shen, J., & An, R. (2025). The application of artificial intelligence techniques in predicting game outcomes of professional basketball league: A systematic review. *PLOS ONE*, 20(6), e0326326. <https://doi.org/10.1371/journal.pone.0326326>
- He, G., Li, C., & Wang, X. (2025). Stacked ensemble model for NBA game outcome prediction. *PMC Articles*.
- Wang, K., & Zemel, R. (2016). Classifying NBA offensive plays using neural networks. In *Proceedings of the MIT Sloan Sports Analytics Conference*. MIT.
- Bianchi, F., Aiello, L. M., & Zignani, M. (2020). Learning player roles from basketball tracking data with graph neural networks. *Machine Learning*, 109(4), 891–911.
- Georgievski, B. (2021). Machine learning and the NBA game. *European Journal of Physical Education and Sport Science*, 8(4), 1–15. <https://efsupit.ro/images/stories/decembrie2021/Art%20453.pdf>